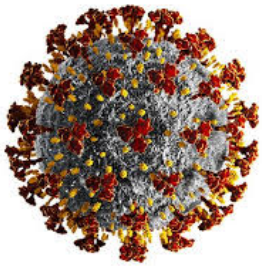
Introduction to Deep Learning - Exercise #1

submission date: 4/11/2021

Programing Task: Antigen Discovery for SARS-CoV-2 ("Corona") Virus Vaccine



Time to end the COVID-19 pandemic by finding potential antigens. The latter are sub-sequences of the virus proteins that can be recognized by our immune system. Our adaptive immune system consists of 6 HLA (class I) alleles that allow it to selectively identify small fragments of proteins, known as peptides. The system is evolved to recognize only peptides of a foreign body and by that invoke an immune proliferation and response of T-cells that destroy the intruder. However, unfortunately not all foreign peptides are recognized. For those of you who are interested in learning more about this mechanism, I suggest this Wiki page.

In this exercise you will train a deep neural network to identify the peptides detected by a specific HLA allele known as HLA_A0201 which is a very common allele shared by 24% of the western population. The training data consists of ~3,000 positive and ~24,500 negative peptides. Each peptide consists of 9 amino acids (of 20 types). At a second stage, you will use your trained predictor to identify sequences of 9-amino-acids peptides from the Spike protein of the SARS-CoV-2 virus.

Formally,

1. You will find the training data at the course's moodle page.
2. Set up a multi-layered perceptron network to accept this data and output the proper prediction (detect / not detect). Try different architectural changes (e.g., different number of levels, neurons at each level, etc.), and non-linearities (RelU, sigmoid) and pick the one achieving the highest accuracy on the test set. Document the tests you conducted in the submission, as well as the best performing architecture.
3. Load the data from the files, and map it to the proper mathematical representation of your choice. Split it into 90%/10% train/test sets (picked at random at each run to avoid over-fitting). Explain your choice of representation in the report. Notice that the training data contains much more negative examples than positive. Use some machine learning strategy to deal with this form of unbalance and avoid convergence to trivial solution such as all negative predictions.
4. Train the network till convergence of the (train/test) loss plots. Make sure your learning rates are not too small and certainly not too large. Detail the chosen parameters in your document and add the train/test loss plots to the submission pdf.
5. Once you find your best configuration, plot the train and test **metrics**, and add it to your submission, in your report explain what methods you have tried.
6. Use your model to predict the detection of 9-amino-acids peptides from the **Spike** protein of the SARS-CoV-2, that is: all its consecutive 9-mer segments out of its

1273-amino-acid sequence. You can download this sequence from this page: https://viralzone.expasy.org/8996

7. Notify the CDC of the 5 most detectable peptides in this protein, as well as include them in your report.
8. Find the most detectable peptides according to your trained network by optimising the networks output score with respect to the input sequence name this function **`optimize_sequence`** . For this purpose you should:
    a. the input tensor should be declared as a trainable variables, by something like:
    ```
    w1 = torch.randn(D_in, H, device=device, dtype=dtype, requires_grad=True)
    ```
    b. And, make sure your optimizer operates on these variables and not your network's weights.
    c. Note that the solution to this problem can be a tensor of arbitrary nature (i.e., have arbitrary values and signs). So consider the way this solution is mapped into a meaningful peptide sequence. Report this resulting sequence.

Theoretical Questions:

1. Show that the composition of linear functions is a linear function. Show that the composition of affine transformations remains an affine function.
2. The calculus behind the Gradient Descent method:
    a. What is the stopping condition of this iterative scheme,

    $$\theta^{n+1} = \theta^n - \alpha \nabla f_{\theta^n}(x)$$

    b. Use the second-order multivariate Taylor theorem,

    $$f(x+dx) = f(x) + \nabla f(x) \cdot dx + dx^T \cdot H(x) \cdot dx + O(\|dx\|^3),$$
    $$H_{ij}(x) = \frac{\partial^2 f}{\partial x_i \partial x_j}(x)$$

    to derive the conditions for classifying a stationary point as local maximum or minimum.
3. Assume the network is required to predict an angle (0-360 degrees). How will you define a prediction loss that accounts for the circularity of this quantity, i.e., the loss between 2 and 360 is not 358, but 2 (since 0 is 360..). Write your answer in a pytorch-compilable form.

4. Chain Rule. Differentiate the following terms (specify the points of evaluation of each function):

a.

$$\frac{\partial}{\partial x} f(x+y, 2x, z)$$

b.

$$f_1\Big(f_2\big(\ldots f_n(x)\big)\Big)$$

c.

$$f_1\Big(x, f_2\big(x, f_3(\ldots f_{n-1}(x, f_n(x)))\big)\Big)$$

d.

$$f\Big(x + g\big(x + h(x)\big)\Big)$$

5. Prove that the Kullback-Libler divergence is non-negative, e.g. $D_{kl}(P||Q) \geq 0 \; s.t \; D_{kl}(P||Q) = 0 \; iff \; P = Q$

6. Prove that the Kullback-Libler divergence ($D_{kl}(P||Q) = \sum_{x \in X} p(x) \cdot log(\frac{p(x)}{q(x)})$) is convex e.g., $D_{kl}(\lambda p_1 + (1-\lambda)p_2 || \lambda q_1 + (1-\lambda)q_2) \leq \lambda \cdot D_{kl}(p_1||q_1) + (1-\lambda) \cdot D_{kl}(p_2||q_2)$ $where \; (p_1, q_1) \; and \; (p_2, q_2) \; are \; two \; pairs \; of \; probability \; distributions \; and \; \lambda \in [0,1]$

7. Explain why Cybenko and Hornik theorems also imply that linear combinations of translated and dilated RelU functions form a dense set in C[0,1].

8. Generalize the construction of a deep network that expresses a shallow network in O(N) neurons, that we saw in class, to signed functions.

Submission Guidelines:

The submission is in **<u>pairs</u>.** Please submit a single zip file named “ex1_ID1_ID2.zip”. This file should contain your code, along with an ”ex1.pdf” file which should contain your answers to the theoretical part as well as the figures/text for the practical part. Furthermore, include in this compress file a README with your names and cse usernames.
Please write readable code, with documentation where needed, as the code will also be checked manually.